Science Advances

AAAS

advances.sciencemag.org/cgi/content/full/3/5/e1602548/DC1

Supplementary Materials for

The ground truth about metadata and community detection in networks

Leto Peel, Daniel B. Larremore, Aaron Clauset

Published 3 May 2017, *Sci. Adv.* **3**, e1602548 (2017)
DOI: 10.1126/sciadv.1602548

This PDF file includes:

Supplementary Text

A The neoSBM

The Supplementary Text is divided into four subsections providing additional details on the neoSBM.

- Subsection I describes the neoSBM (I.a) and the inference methods used in this paper (I.b).
- Subsection II describes the generation of the synthetic network used in the main text, Fig. 3.
- Subsection III describes how the neoSBM can be extended to other models including the degree corrected neoSBM.
- Subsection IV provides additional examples of results of the neoSBM applied to the Lazega Lawyers networks (IV.a) and the Malaria networks (IV.b).

For convenience, we provide a reference table of notation used in derivations in the Supplementary Text.

table S1. Notation used in the Supplementary Text.

Variable	Definition
$\mathcal{G}$	a network, $\mathcal{G} = (V, E)$
N	the number of nodes $\|V\|$
A_{ij}	the number of edges between nodes i and j, $A_{ij} \in \{0, 1\}$
k_i	the degree of node i.
ω_{rs}	the probability of an edge between nodes in groups r and s
π	a partition of nodes into groups
M	a set of metadata labels
C	an inferred optimal community assignment
z	neo-state indicator variable, $z_i \in \{b, r\}$
θ	Bernoulli prior probability parameter
$\mathcal{L}_X$	log likelihood L of model X
q	the number of free nodes, $q = \sum_i \delta_{z_i,r}$
$\delta_{a,b}$	the Kronecker delta: $\delta_{a,b} = 1$ for $a = b$; $\delta_{a,b} = 0$ for $a \neq b$

A.1 neoSBM model description and inference

A.1.1 Model description

The neoSBM extends the SBM, allowing metadata to influence the inferred partitions by controlling the number of nodes that are assigned to groups according to their metadata labels. The task of the neoSBM is to perform community detection under a constraint in which each node is assigned a latent state variable z_i, which can take one of two states, which we call blue or red. If a node is blue $z_i = b$, its community is fixed as its metadata label $\pi_i = M_i$. However, if it is red $z_i = r$, its community is free to be chosen by the model. We adjust the number of free nodes q by varying the Bernoulli prior probability θ that a node will be free (red state). We can then write down the likelihood L_{neo} of a network $\mathcal{G}$ given a community assignment π under the neoSBM as

$$L_{\text{neo}}(\mathcal{G};\pi,z) = \prod_{ij} \omega_{\pi_i\pi_j}^{A_{ij}}(1-\omega_{\pi_i\pi_j})^{(1-A_{ij})} \prod_i \theta^{\delta_{z_i,r}}(1-\theta)^{\delta_{z_i,b}} \tag{5}$$

The first product in Eq. (5) corresponds to the standard SBM likelihood L_{sbm}, while the second product corresponds to the probability of the states $P(z = r|\theta)$ and acts as a penalty function to control the number of free nodes. While it is possible to find communities by optimizing Eq. (5) directly, instead we work with the more practical log likelihood

$$\begin{aligned}\mathcal{L}_{\text{neo}}(\mathcal{G};\pi,z) = &\sum_{ij} A_{ij}\log\omega_{\pi_i\pi_j} + (1-A_{ij})\log(1-\omega_{\pi_i\pi_j}) \\ &+ \sum_i \delta_{z_i,r}\log\theta + \delta_{z_i,b}\log(1-\theta)\end{aligned} \tag{6}$$

since maximizing Eq. (5) is equivalent to maximizing Eq. (6). We can then rearrange the second sum $\log P(z = r|\theta)$, to give

$$\begin{aligned}\log P(z=r|\theta) &= \sum_i \delta_{z_i,r}\left(\log\frac{\theta}{1-\theta}\right) + N\log(1-\theta) \\ &= q\psi(\theta) + N\log(1-\theta)\end{aligned} \tag{7}$$

dropping the constant term, we can rewrite the neoSBM log likelihood in terms of the SBM log likelihood and a function of the number of free nodes q

$$\mathcal{L}_{\text{neo}}(\mathcal{G}; \pi, z) = \mathcal{L}_{\text{sbm}}(\mathcal{G}; \pi) + q\psi(\theta) \tag{8}$$

We emphasize that in the equation above, θ is a fixed parameter, and q is selected automatically during inference as part of the likelihood maximization. Optimization of $\mathcal{L}_{\text{sbm}}$ yields the SBM optimal communities C

$$C = \arg\max_{\pi} \mathcal{L}_{\text{sbm}}(\mathcal{G}; \pi) \tag{9}$$

and so the SBM likelihood given the metadata partition M will always be less than or equal to the likelihood of the inferred partition C. That is $\mathcal{L}_{\text{sbm}}(\mathcal{G}; M) \leq \mathcal{L}_{\text{sbm}}(\mathcal{G}; C)$, where the inequality is saturated if and only if the metadata is equal to the optimal SBM partition. So the minimum number of free nodes $\hat{q}$ required to maximize the SBM likelihood is

$$\hat{q} = \sum_{i} 1 - \delta_{M_i, C_i} \tag{10}$$

for which the label permutations of M and C are maximally aligned. Whenever $q > \hat{q}$ there will be no further improvement in $\mathcal{L}_{\text{sbm}}$. To interpolate between M and C we vary the prior probability of each node to take the red state $P(z = r|\theta)$. For values of $\theta < 0.5$ we can interpret the log probability, or $\psi(\theta)$, as the cost of freeing a node because the log likelihood $\mathcal{L}_{\text{neo}}$ will incur a penalty for setting each $z_i = r$. Maximizing $\mathcal{L}_{\text{neo}}$ is therefore a trade-off between freeing nodes to maximize $\mathcal{L}_{\text{sbm}}$ and fixing nodes to metadata labels to maximize $\log P(z|\theta)$. When the SBM likelihood of both partitions is equal (i.e., $M = C$) then $\mathcal{L}_{\text{neo}}(\mathcal{G}; \pi, z)$ will be maximized when $q = 0$ unless $\theta \geq 0.5$. However, when $\mathcal{L}_{\text{sbm}}(\mathcal{G}; M) < \mathcal{L}_{\text{sbm}}(\mathcal{G}; C)$, q can be greater than 0 if the resulting partition π provides a sufficient increase in log likelihood. Specifically, if

$$\mathcal{L}_{\text{sbm}}(\mathcal{G}; \pi) - \mathcal{L}_{\text{sbm}}(\mathcal{G}; M) > q\psi(\theta) \tag{11}$$

then it indicates that the cost of freeing q nodes is outweighed by its contribution to improving the likelihood.

Here we have discussed the extension of the SBM to the neoSBM, but this extension can be easily generalized to any probabilistic generative network model that specifies the likelihood of a graph given a partition of the network. We present one such generalization, the degree-corrected neoSBM, in subsection III of the Supplementary Text.

A.1.2 Inference

Inference of the parameters of the neoSBM was performed using a Markov chain Monte Carlo (MCMC) approach. The community labels of the free nodes were inferred in the same way as the standard SBM (*53*). However, to infer the values of z_i that determined whether or not each node was free, we used a uniform Bernoulli (i.e., a fair coin) as a proposal distribution. Since this distribution is symmetric we can simply accept each proposal with probability a

$$a = \min\left\{\Delta L_{\text{neo}}, 1\right\} \tag{12}$$

To avoid getting trapped in local optima of the likelihood, we initialize the neoSBM with the labels set to the inferred SBM partition, $\pi = C$, and all nodes initialized to be free, $z_i = r$ for all i.

A.2 Extensions

The neoSBM can easily be extended to any probabilistic model for which we identify communities by maximizing the model likelihood. As an example, consider the degree-corrected SBM, which allows for nodes with heterogenous degrees to belong to the same community (see Supplementary Text B for more details). We can create a degree-corrected neoSBM in much the same way as we created the neoSBM, by penalizing the likelihood according to the number

of free nodes using a Bernoulli prior. This treatment gives the log likelihood

$$\mathcal{L}_{\mathrm{dcneo}}(\mathcal{G};\pi,z) = \mathcal{L}_{\mathrm{dcsbm}}(\mathcal{G};\pi) + q\psi(\theta) \tag{13}$$

where $q\psi(\theta) = q\log P(z = r|\theta) + N\log(1-\theta)$ as before. We present results from this model in subsection IV of the Supplementary Text.

We can also easily extend the neoSBM to other, non-probabilistic, community detection methods provided they explicitly optimize a global objective function. Then we can similarly create a penalized version of this objective function. That is, for some community detection model X, we can create a *neo*-objective function $\mathcal{U}_{neoX}$

$$\mathcal{U}_{neoX} = \mathcal{U}_X + q\psi(\theta) \tag{14}$$

where $\psi(\theta)$ could either represent the Bernoulli prior as before or any other cost function, e.g., $\psi(\theta) = \theta$, for $\theta \leq 0$.

A.3 IV. Results on real-world networks

In order to further demonstrate the neoSBM and the neoDCSBM described above, we present and discuss the application of the neoSBM to malaria *var* gene networks and the application of the neoDCSBM to the Karate Club network. Full details about these data sets are presented in Supplementary Text D.

A.3.1 neoDCSBM and the Karate Club network

The likelihood surface for both models contains two local optima that correspond two the same two partitions, each being globally optimal for one of the models. Using the faction each member joined after the club split as metadata fig. S1 compares the output from the neoSBM and the neoDCSBM. Both models initially change just a single node to reach a local optimum. For the DCSBM this is the global optimum and so we see no further change. However, for the neoSBM

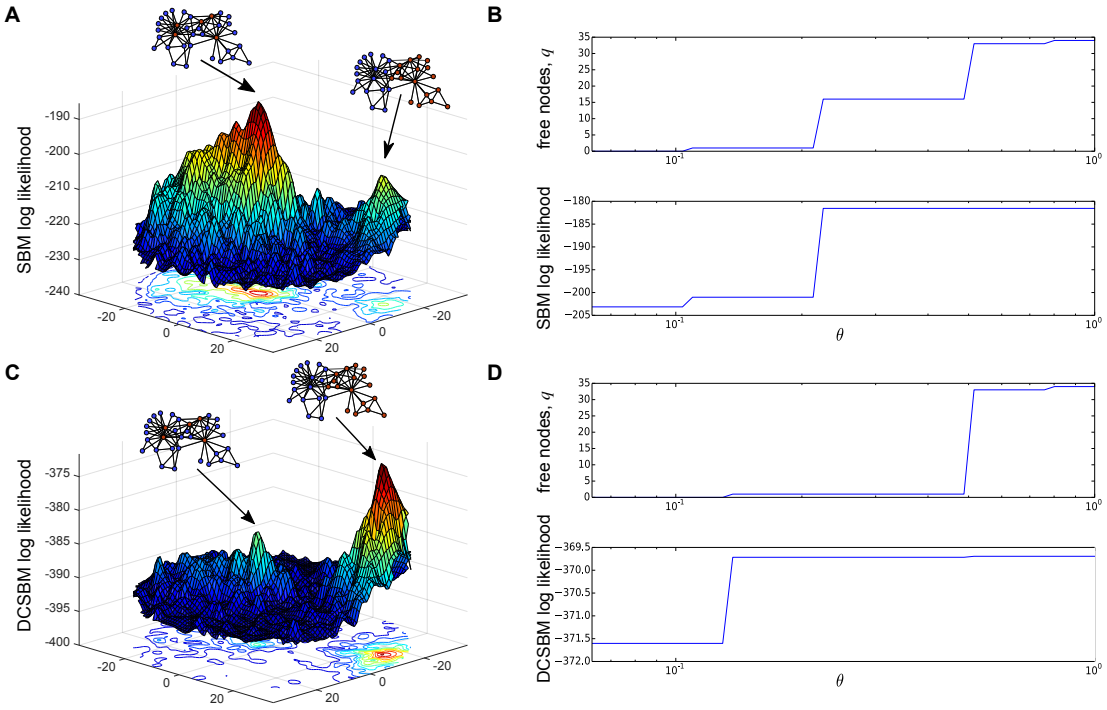

fig. S1. The results of the neoSBM and the degree-corrected neoSBM on the Karate Club network. The SBM and DCSBM log likelihood surfaces (A and C respectively) show distinct two peaks that correspond to the same two partitions of the network: the two social factions and the leader-follower partition. When we use the faction partition as metadata, we from the output (B and D) that both models change a single node in order to reach the locally optimal partition. For the neoDCSBM (D), this is the global optimum and no further change is observed. For the neoSBM, the leader-follower partition is globally optimal, so once theta is large enough we see the model jump to this partition.

this is not the global optimum (see Fig. 1) and so once θ is large enough we see a discontinuous jump as it switches to the globally optimal high-degree/low-degree partition.

A.3.2 neoSBM and the Malaria *var* gene networks

The metadata corresponding to upstream promoter sequence (UPS) are known to correlate with community structure in the malaria *var* gene networks, particularly at loci one and six (*21, 41*). We provided the neoSBM with UPS metadata ($K = 4$) and investigated the path of partitions between the metadata partition and the globally optimal partitions for each of the two networks.

Figures S3 (locus one) and S4 (locus six) show likelihood surfaces, block density diagrams, and the neoSBM's outputs q (free nodes) and SBM log likelihood.

Comparison of the neoSBM results for the same metadata on two different network layers reveals not only that the intermediate paths of locally optimal partitions differ but that the UPS metadata are more locally stable for the locus six network. This is indicated by the substantially larger value of θ at which the neoSBM switches from the metadata partition to the first intermediate local optimum. These transitions $1 \rightarrow 2$ involve different numbers of free nodes, however, indicating that the switch from optimum 1 to optimum 2 was accompanied by a much larger change in node mobility for the locus six network. Note that the neoSBM provides a more nuanced view of the relationship between UPS metadata and malaria layers one and six than the BESTest did, which found that UPS metadata were significantly correlated with the structures of both networks.

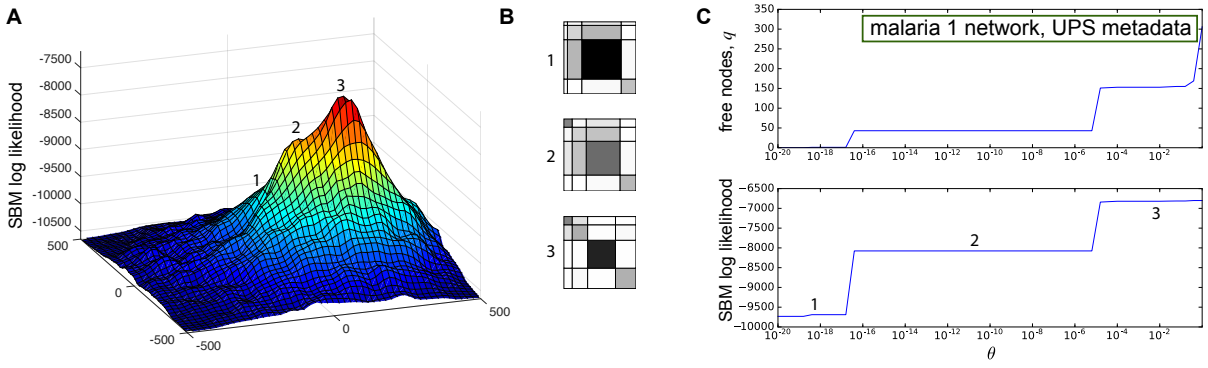

fig. S2. Results of the neoSBM on the malaria *var* gene network at locus one ("malaria 1") using UPS metadata. (A) The SBM likelihood surface shows two peaks, one subtle 2 and one prominent 3, corresponding to a locally optimal partition near the metadata and the globally optimal partition, respectively. There is no peak at the metadata partition 1, however. (B) Block density diagrams depict community structure for metadata and locally optimal partitions, where darker color indicates higher probability of interaction. (C) The neoSBM, beginning from UPS metadata, interpolates between metadata 1 and the globally optimal SBM partition 3. The number of free nodes q and SBM log likelihood as a function of θ shows two discontinuous jumps as the neoSBM traverses from the metadata to the locally optimal partition ($1 \rightarrow 2$) and then from that partition to the global optimum ($2 \rightarrow 3$).

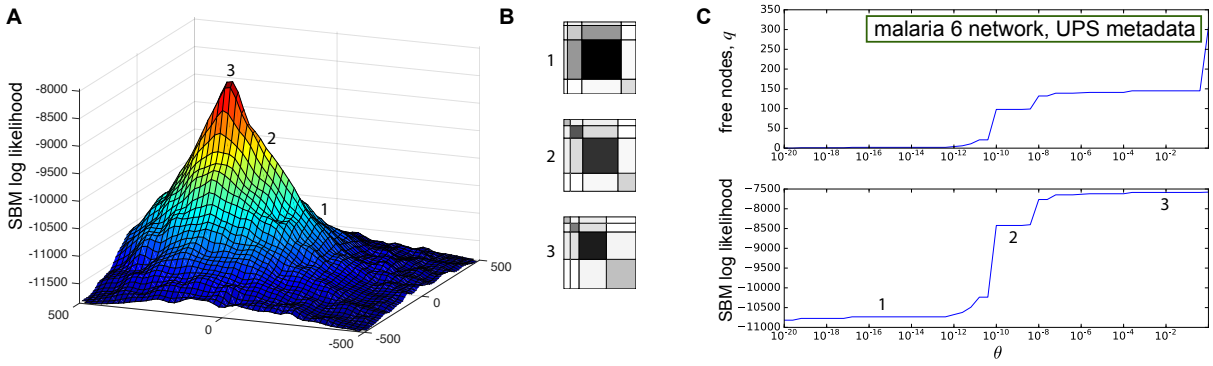

fig. S3. Results of the neoSBM on the malaria *var* gene network at locus six ("malaria 6") using UPS metadata. (A) The SBM likelihood surface shows one prominent peak at the globally optimal partition. (B) Block density diagrams depict community structure for metadata and locally optimal partitions where darker color indicates higher probability of interaction. (C) The neoSBM, beginning from UPS metadata, interpolates between metadata 1 and the globally optimal SBM partition, traversing a local optimum during its path. The number of free nodes q and SBM log likelihood as a function of θ shows two discontinuous jumps as the neoSBM traverses from the metadata to the locally optimal partition ($1 \rightarrow 2$), from that partition to another the global optimum ($2 \rightarrow 3$).

A.4 Synthetic network generation for the neoSBM

The test that demonstrated the function of the neoSBM on synthetic data, depicted in Fig. 3 of the main text, required networks with multiple local optima under the SBM: one corresponding to the inferred partition (global optimum) and at least one other to represent a relevant metadata partition. To create such a network, we divided vertices into $2K$ groups to create K assortative communities, each of which was subdivided to contain a core and a periphery group. For $K = 4$, fig. S5 shows the 8-block interaction matrix used to create the synthetic networks. By subsequently varying the mean degree within each block, we obtained two uncorrelated partitions when $K = 4$, both of which are relevant to the network structure. Finally, we assigned as metadata the core-periphery structure containing one periphery group ($\{2, 4, 5, 7\}$ in fig. S5) and three core groups ($\{1,3\}$,$\{6\}$,$\{8\}$ in fig. S5). The partition inferred by the SBM in the absence of the neoSBM's likelihood penalty corresponds to the assortative group structure.

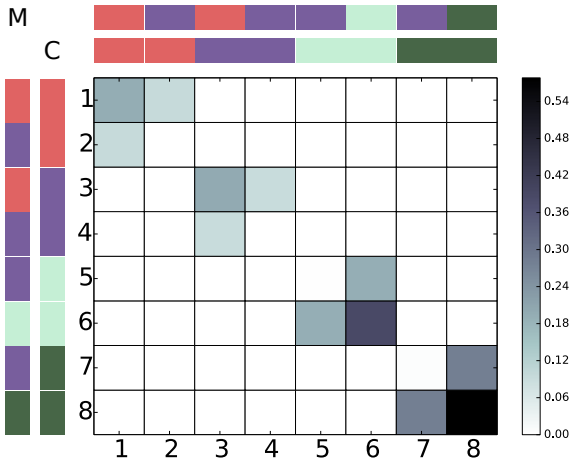

fig. S4. The block interaction matrix used to generate synthetic networks. The external colored rows and columns indicate the partition used as metadata (M) and the maximum likelihood partition under the SBM (C).

B Blockmodel Entropy Significance Test

The Supplementary Text is divided into six subsections providing additional details on the blockmodel entropy significance test.

- Subsection B.1 describes maximum likelihood parameter estimation for the SBM (I.a) and degree-corrected SBM (I.b).
- Subsection B.2 describes rapid computation of the entropy $H(\mathcal{G};\mathcal{M})$ for the Bernoulli SBM and Multinomial degree-corrected SBM (DCSBM).
- Subsection B.3 demonstrates the mathematical link between our formulation of the SBM entropy and the SBM log likelihood which has been derived elsewhere (*20, 54*).
- Subsection B.3 discusses the use of non-generative models like modularity.
- Subsection B.5 provides details on the generation of synthetic networks for the tests shown in Fig. 2.
- Subsection B.6 provides additional examples of results of the blockmodel entropy significance test using multiple different network data and metadata sets (see Supplementary Text D) as well as three additional generative network models beyond the SBM.

For convenience, we provide a reference table of notation used in derivations in the Supplementary Text.

B.1 Estimation of SBM parameters

B.1.1 Bernoulli SBM parameters

Let the N nodes of a network $\mathcal{G}$ be partitioned into K groups, with the group assignment of node i given by π_i. In the SBM, the probability of a link existing between any two nodes i and

table S2. Notation used in the Supplementary Text.

Variable	Definition
$\mathcal{G}$	a network, $\mathcal{G} = (V, E)$
N	the number of nodes $\|V\|$
π	a partition of nodes into groups
K	the total number of groups
π_i	the group assignment of node i
n_r	the number of nodes in group r
m_{rs}	the number of edges between groups r and s
κ_r	the total degrees of group r, $\kappa_r = \sum_s m_{rs}$
k_i	the degree of node i.
$H_X(\mathcal{G}\|\pi)$	entropy H of model X estimated for graph $\mathcal{G}$ using partition π
$\hat{a}$	maximum likelihood estimate of model parameter a
p_{ij}	the probability that an edge exists between nodes i and j

j depends only on the group assignments π_i and π_j. This means that the entire model can be parameterized by a $K \times K$ matrix of block-to-block edge probabilities, ω. Accordingly, let ω be a matrix such that $p_{ij} = \omega_{\pi_i \pi_j}$ is the probability of a link existing between i and j. Letting the number of nodes in group r be n_r, then between two groups r and s there are $n_r n_s$ possible links, each of which has the same probability of existence, ω_{rs}. This implies that the existence of the $n_r n_s$ edges between groups r and s will be determined by $n_r n_s$ independent Bernoulli trials, each with parameter ω_{rs}.

We must now estimate the value of ω_{rs} for a network $\mathcal{G}$ whose nodes have been divided according to their assignments in partition π. Of course, any ω whose entries are positive will have some non-zero probability of having generated the observed links in $\mathcal{G}$. However, here we choose the values of ω to be those that maximize the likelihood of observing $\mathcal{G}$. Specifically, observe that of the $n_r n_s$ Bernoulli trials, there are m_{rs} actual edges in the graph, i.e., m_{rs} trial successes. Therefore, the maximum likelihood estimate of ω_{rs} is simply $\hat{\omega}_{rs} = m_{rs}/n_r n_s$. Thus, $\hat{p}_{ij} = \hat{\omega}_{\pi_i \pi_j}$.

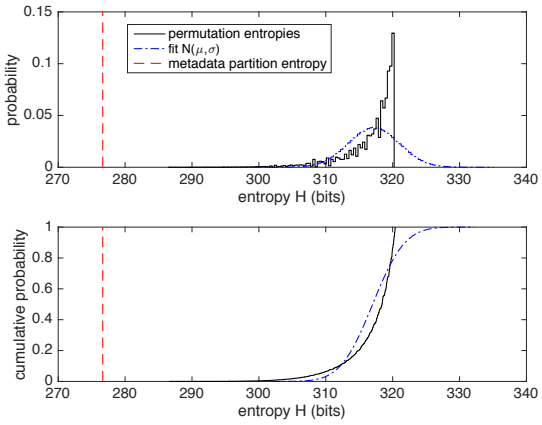

fig. S5. Distributions of permuted partition entropies are negatively skewed. Probability density functions (top) and cumulative distribution functions (bottom) are shown for the entropies of partitions of the Karate Club network and its faction metadata. The red broken line indicates the point entropy of the metadata partition while the black solid line shows the distribution of entropies for 10^4 independent permutations of the metadata partition. Note that these permutation entropies are far from normal; a normal distribution with equivalent mean μ and variance σ^2 is shown in blue for contrast.

B.1.2 Poisson degree-corrected SBM parameters

In the degree-corrected Poisson SBM (*20*), it is still assumed that each link exists independently of the others, with some specified probability given by a block connectivity matrix ω. However, this model differs in two key ways from the Bernoulli SBM. First, rather than each edge existing with probability p_{ij}, Poisson SBMs state that the *expected* number of edges between nodes i and j is given by a parameter q_{ij}, with the actual number of edges drawn from a Poisson distribution with identical mean. For very small values of q, the probability of an edge existing is approximately q, and thus if the graph is sufficiently sparse, Poisson SBMs behave similarly to Bernoulli SBMs, despite the fact that they could, in principle, generate multigraphs.

The second way in which this degree-corrected Poisson SBM differs from the Bernoulli SBM is that the parameters q_{ij} are no longer identical across the set of all i in group r and all j in group s, as they are in the uncorrected SBM. Now, each node has a degree affinity θ_i so

that $q_{ij} = \theta_i \theta_j e_{\pi_i \pi_j}$, where e_{rs} is the $K \times K$ block structure matrix, controlling the numbers of links between groups, similar in principle to ω_{rs} above. The new parameters, θ_i, properly chosen (*20*), can be used to specify the expected degree of each node.

As above, since we are given a network $\mathcal{G}$ and a fixed partition π, we must estimate the entries of e, as well as the values of θ. The parameters can again be chosen to maximize the likelihood of observing $\mathcal{G}$, which are derived in (*20*) but we do not derive here. First, $\hat{e}_{rs} = m_{rs}$, where $m_{rs} = \sum_{ij} A_{ij} \delta_{r,\pi_i} \delta_{s,\pi_j}$ is the number of links between groups r and s (or twice the number of links if $r = s$). Then, $\hat{\theta}_i = k_i / \kappa_{\pi_i}$, where κ_r is the number of degrees connecting to group r, $\kappa_r = \sum_s m_{rs}$. Thus, $\hat{q}_{ij} = k_i k_j m_{\pi_i \pi_j} / \kappa_{\pi_i} \kappa_{\pi_j}$. We note that this maximum likelihood estimate is only valid in the regime that $k_i k_j m_{\pi_i \pi_j} \ll \kappa_{\pi_i} \kappa_{\pi_j}$.

B.2 Rapidly computing entropy

B.2.1 Rapid Bernoulli SBM entropy

Under either a Bernoulli-type SBM, a link exists between nodes i and j with probability p_{ij}, independently of all other links. This amounts to a Bernoulli trial or flip of a biased coin, and the entropy of this Bernoulli trial with parameter p_{ij} is simply

$$h(p_{ij}) \equiv -p_{ij} \log_2 p_{ij} - (1 - p_{ij}) \log_2 (1 - p_{ij}) \tag{15}$$

Hereafter, we will write simply $\log$ in place of $\log_2$. Because the Bernoulli trial on each link is conditionally independent of other links, the entropy of the network is the sum of all valid $h(p_{ij})$. For an undirected network this is

$$H_{\text{SBM}}(\mathcal{G}) = \sum_{i \leq j} h(p_{ij}) = \frac{1}{2} \left[\sum_{ij} h(p_{ij}) + \sum_{i} h(p_{ii}) \right] \tag{16}$$

Under the SBM, the probabilities within each block are identical so we may group them and change to an index over groups, rewriting Eq. (16) as

$$H_{\text{SBM}}(\mathcal{G}) = \frac{1}{2} \left[\sum_{rs} n_r n_s h(\omega_{rs}) + \sum_{r} n_r h(\omega_{rr}) \right] \tag{17}$$

which may be simplified by plugging in the maximum likelihood estimate of $\hat{\omega}_{rs}$ and the definition of Bernoulli entropy h Eq. (15), yielding

$$H_{\text{SBM}}(\mathcal{G}) = \dots$$
$$-\frac{1}{2}\left[\sum_{rs} m_{rs}\log\hat{\omega}_{rs} + (n_r n_s - m_{rs})\log(1-\hat{\omega}_{rs})\right] + \mathcal{O}(n^{-1}) \tag{18}$$

where we have noted that the diagonal terms are $\mathcal{O}(n^{-1})$ whenever $n_r = cn$ for some constant c.

Eq. (18) allows for a $\mathcal{O}(K^2)$ computation, rather than $\mathcal{O}(N^2)$ of Eq. (16). For degree-corrected Bernoulli SBMs, entropies may be summed as in Eq. (16), even though the rapid computation of Eq. (18) will not be valid. However, in what follows, we show the connection between model entropy H and model log likelihood $\mathcal{L}$.

B.2.2 Rapid Multinomial DCSBM entropy

The degree-corrected SBM, introduced as a Poisson DCSBM by Karrer and Newman (*20*), can also be written in a "Multinomial" form in which each of the m edges is placed sequentially, according to the multinomial probabilities p_{ij} (*55*). The values of p_{ij} are defined as

$$p_{ij} = \theta_{ir}\omega_{rs}\theta_{js} = \frac{k_i k_j e_{rs}}{2m e_r e_s} \tag{19}$$

where $\theta_{ir} = k_i/e_r$ if node i is in group r, and 0 otherwise, and $\omega_{rs} = e_{rs}/2m$. Note that by definition, $\sum_{ij} p_{ij} = 1$. When constructing a network, m edges are placed among the possible edge locations, with each one independently according to a categorical distribution with probabilities p_{ij} (*55*).

Since it is possible that multiple edges are formed between pairs of vertices, the entropy of this ensemble is not the entropy of m categorical distributions with parameters p, but rather

the entropy of the multinomial distribution with m draws and b "bins" with parameters p. Note that if there is a nonzero possibility of an edge between each pair of vertices, then $b = \binom{N}{2}$ [or $b = N^2$ in the directed case]. (There may be fewer than $\binom{N}{2}$ bins in the undirected case if some values of e_{rs} are equal to 0, and similarly, there may be fewer than N^2 bins in the directed case if some values of ers, k_i^{out}, or k_i^{in} are equal to 0.) There is no closed-form expression for the entropy of a multinomial distribution but an accurate approximation has been derived in (*56*), into which we substitute the parameters of the multinomial DSCBM, yielding

$$H = \frac{1}{2}\log\left[(2\pi me)^{b-1}\prod_{ij;p_{ij}\neq 0} p_{ij}\right] + \frac{1}{12m}\left[3b - 2 - \sum_{ij;p_{ij}\neq 0}\frac{1}{p_{ij}}\right] + \mathcal{O}\left(\frac{1}{m^2}\right) \tag{20}$$

Thus, computing the entropy of this degree-corrected model (*55*) amounts to the rapid estimation of the parameters p from Eq. (19) followed by computation of the entropy from Eq. (20).

B.3 Connecting entropy and log likelihood

The connection between model entropy H and model log likelihood $\mathcal{L}$ enables the blockmodel entropy significance test to be expanded beyond the simple Bernoulli SBM to degree-corrected SBMs, Poisson SBMs, mixed-membership models, and other generative models with computable log likelihoods.

We begin from Eq. (18) and use the Taylor series

$$(1-x)\ln(1-x) = -x + \sum_{\ell=2}^{\infty}\frac{x^\ell}{\ell(\ell-1)} \tag{21}$$

in which we substitute $x = \hat{\omega}_{rs} = m_{rs}/n_r n_s$ to write Eq. (18) to leading order as

$$H_{\text{SBM}}(\mathcal{G}) \approx -\frac{1}{2}\sum_{rs}\left[m_{rs}\ln\left(\frac{m_{rs}}{n_r n_s}\right) - m_{rs}\ldots + n_r n_s\sum_{\ell=2}^{\infty}\frac{1}{\ell(\ell-1)}\left(\frac{m_{rs}}{n_r n_s}\right)^\ell\right] \tag{22}$$

Finally, we note that $\frac{1}{2}\sum_{rs} m_{rs}$ is simply $|E|$, the total number of links in the network and therefore

$$H_{\text{SBM}}(\mathcal{G}) \approx |E| - \frac{1}{2}\sum_{rs}\left[m_{rs}\ln\left(\frac{m_{rs}}{n_r n_s}\right)\ldots + n_r n_s \sum_{\ell=2}^{\infty}\frac{1}{\ell(\ell-1)}\left(\frac{m_{rs}}{n_r n_s}\right)^{\ell}\right] \tag{23}$$

If all blocks of links are sparse, then $m_{rs} \ll n_r n_s$ and the terms in the infinite sum decay rapidly, leading to the first order approximation

$$H_{\text{SBM}}(\mathcal{G}) \approx |E| - \frac{1}{2}\sum_{rs} m_{rs}\ln\left(\frac{m_{rs}}{n_r n_s}\right) \tag{24}$$

Here we derived Eq. (23) and Eq. (24) by considering the conditionally independent entropies associated with every link of the network. However, the same equations can also be derived by calculating the size Ω of the ensemble of networks associated with the same SBM, and then taking a logarithm, $H = \log\,\Omega$ The log likelihood is the logarithm of the probability of observing an individual network realization from the ensemble, $\mathcal{L} = \log P$, and under the assumption that each graph in the ensemble occurs with the same probability, $P = 1/\Omega$. Therefore, the entropy H and the log likelihood $\mathcal{L}$ are related simply by $\mathcal{L} = -H$ (*54*).

The relationship between the "microcanonical" entropy and log likelihood allows for the Blockmodel Entropy Significance test to be expanded easily to any generative model for networks for which a likelihood is easily computed,

$$p\text{-value} = \Pr\left[\mathcal{L}(\mathcal{G};\tilde{\pi}) \geq \mathcal{L}(\mathcal{G};\mathcal{M})\right] \tag{25}$$

The Bernoulli SBM entropy Eq. (18) or its approximation for sparse networks Eq. (24) are convenient because they are fast to compute—one need only to count links between groups, sizes of groups, and compute $\mathcal{O}(K^2)$ terms. By contrast, Eq. (16), which is exact, requires $\mathcal{O}(N^2)$ computations. Depending on the assumptions involved, computing a log likelihood $\mathcal{L}$ may be more or less rapid, or more or less exact.

In the additional tests in the Supplementary Text, we employ the Likelihood equations to apply the BESTest using the Poisson SBM and degree-corrected SBM, and use the rapid entropy equations for the Bernoulli SBM and Multinomial DCSBM.

Finally, we note that an alternative version of entropy that is not based on the blockmodel but instead by the size of the ensemble of networks with identical degree sequence and communities is discussed in Ref. (*40*).

B.4 Application of the significance test approach to non-generative models for community structure

The blockmodel entropy significance test provides an estimate of how often a given partition provides a lower-entropy explanation of the data, as viewed through a particular model. While we have, so far, derived expressions for this test in terms of the entropy of a model Eq. (4) or its likelihood Eq. (25), there exist many other approaches to community detection that are not generative, and therefore have neither a likelihood or an entropy. These models rely on a quality function or Hamiltonian which is optimized over partitions. Supposing that optimization of the Hamiltonian $\mathcal{Q}$ involves maximization, the test statistic is

$$p\text{-value} = \Pr\left[\mathcal{Q}(\mathcal{G};\tilde{\pi}) \geq \mathcal{Q}(\mathcal{G};\mathcal{M})\right] \tag{26}$$

If optimization involves minimization of $\mathcal{Q}$, the direction of the inequality above should be reversed.

Modularity (*9*), one of the most popular quality functions used for community detection, serves as an instructive example of the blockmodel entropy significance test in two ways. First, it is a measure of the strength of the assortment of links into communities, but has no generative model. Indeed, sampling from the space of networks with a particular modularity NP-hard (*57*). Second, the modularity score itself defines community structure narrowly as assortative, and

therefore networks with disassortative structures, which have significant p-values using any SBM as the test model, are likely to be found to have non-significant p-values when Eq. (26) is used. This emphasizes both the versatility of the test statistic, as well as the differences between definitions of community structure—spanning generative and non-generative models alike.

It is worth noting that the value of modularity is asymptotically zero whenever assortative communities are uncorrelated with the partition at which it is being evaluated—indeed, the premise of modularity maximization is to find communities whose internal edges defy expectation based on this uncorrelated null model. Thus, if metadata provide a partition of the network, and modularity is found to be exactly zero, then from the perspective of the particular type of assortative structure defined by modularity, there is not a significant relationship between metadata and community structure. On the other hand, simply finding that modularity under a particular metadata partition $\mathcal{Q}(\mathcal{G}, \mathcal{M})$ is non-zero need not imply that the relationship is or is not statistically significant in the sense of Eq. (26); the test must be performed.

B.5 Generation of synthetic networks for blockmodel entropy significance test

The tests described in the main text, and detailed in the Supplementary Text, will yield a p-value which indicates the extent to which a set of metadata (and a generative model) describes a network better than a random partition. In order to understand the sensitivity of the BESTest, we generated sets of synthetic networks and synthetic metadata, applied the BESTest to them, and produced Fig. 2. Here we describe the process used to generated those synthetic networks.

We generated networks of $N = 1000$ nodes and two planted communities r and s using the (Bernoulli) SBM. Each node was assigned to one of the communities ($\mathcal{T}_i = r$ or $\mathcal{T}_i = s$) with equal probability. We then generated a network with a given community strength $\epsilon = \omega_{rs}/\omega_{rr}$ such that low values of ϵ generate strongly assortative communities with few connecting edges between them and as ϵ grows, the generated communities become weaker, producing a random

table S3. Lazega Lawyers: BESTest P values.

Network	Attribute				
	Status	Gender	Office	Practice	Law School
			SBM		
Friendship	$< 10^{-6}$	0.034	$< 10^{-6}$	0.033	0.134
Cowork	$< 10^{-3}$	0.094	$< 10^{-6}$	$< 10^{-6}$	0.922
Advice	$< 10^{-6}$	0.010	$< 10^{-6}$	$< 10^{-6}$	0.205
			DCSBM		
Friendship	$< 10^{-6}$	0.001	$< 10^{-6}$	0.002	0.094
Cowork	$< 10^{-6}$	0.842	$< 10^{-6}$	$< 10^{-6}$	0.938
Advice	$< 10^{-6}$	0.205	$< 10^{-6}$	$< 10^{-6}$	0.328
			Poisson SBM		
Friendship	$< 10^{-6}$	0.046	$< 10^{-6}$	0.044	0.167
Cowork	$< 10^{-3}$	0.099	$< 10^{-6}$	$< 10^{-6}$	0.977
Advice	$< 10^{-6}$	0.013	$< 10^{-6}$	$< 10^{-6}$	0.316
			Poisson DCSBM		
Friendship	$< 10^{-6}$	$< 10^{-3}$	$< 10^{-6}$	$< 10^{-3}$	0.014
Cowork	$< 10^{-4}$	0.969	$< 10^{-6}$	$< 10^{-6}$	0.781
Advice	$< 10^{-5}$	0.018	$< 10^{-6}$	$< 10^{-6}$	0.046

graph with no communities when $\epsilon = 1$. For each node i, with probability ℓ we assigned its metadata label to be its community label ($\mathcal{M}_i = \mathcal{T}_i$), otherwise we assigned it a uniformly random label. Thus, as ℓ increases from 0 to 1 the metadata labels correlate more with the planted communities, and the probability that any individual node's metadata label matches its community label is $\ell(1) + (1 - \ell)(1/2) = (1 + \ell)/2$.

B.6 Additional applications of the BESTest to real data

We now present and discuss the results of applying the BESTest to the Lazega Lawyers and Malaria data sets (see Supplementary Text D).

B.6.1 Lazega Lawyers

We applied the BESTest to all three Lazega Lawyers networks (Friendship, Cowork, Advice) which share the same set of nodes but have different sets of edges, representing different relationships between individuals. There were five sets of node metadata (Status, Gender, Office, Practice, and Law School). We applied the BESTest to each combination of network and metadata, using four generative models (SBM, degree-corrected SBM, Poisson SBM, and Poisson degree-corrected SBM). These results are shown in table S3.

First, note that values between Bernoulli and Poisson models are not identical, though they are similar, implying that the models are not entirely interchangeable. More importantly, however, the results for degree-corrected and degree-uncorrected models are substantially more different, with relationships varying from significant under one model to insignificant under another. This highlights the fact that metadata can explain patterns of group structure in a network only through the lens of a particular network generative model; a change in the model may impact the metadata's ability to explain patterns in network community structure.

Second, note that under all models, for each network there exist multiple sets of metadata that are significant. Similarly, there exist multiple networks for which any individual set of metadata is significant. This fundamentally undermines the notion that one should expect a single set of metadata to function as ground truth, given that multiple sets of metadata explain multiple networks.

B.6.2 Malaria

We applied the BESTest to nine layers of a network of malaria parasite genes (Malaria 1-9) using four generative models (SBM, degree-corrected SBM, Poisson SBM, and Poisson degree-corrected SBM). Three sets of metadata exist for these networks, (parasite origin, CP group, and UPS), described in detail in Supplementary Text D.

table S4. Malaria: BESTest P values for parasite origin metadata.

	Model			
Network	SBM	DCSBM	Poi. SBM	Poi. DCSBM
Malaria 1	0.566	0.096	0.606	0.086
Malaria 2	0.064	0.148	0.066	0.143
Malaria 3	0.536	0.389	0.532	0.501
Malaria 4	0.588	0.617	0.604	0.644
Malaria 5	0.382	0.077	0.369	0.087
Malaria 6	0.275	0.923	0.293	0.751
Malaria 7	0.020	0.388	0.019	0.501
Malaria 8	0.464	0.176	0.468	0.172
Malaria 9	0.115	0.067	0.108	0.200

The *parasite origin* results are shown in table S4, and none of the p-values listed is significant. This result indicates that when the nodes of each layer are divided into groups based on parasite origin, the entropy of the resulting model is no better than assigning the nodes to groups at random. This implies, in turn, that the malaria parasite antigen genes do not group by the parasite from which they came, confirming previous observations (*41*). However, as shown in Fig. 2 the BESTest is sensitive to even small levels of explanatory power provided by metadata, indicating that parasite origin has truly no bearing on the community structure of malaria parasite antigen genes, for all four generative models tested.

On the other hand, it is known that the genes represented by the nodes of the malaria parasite networks are correlated with CP group and UPS metadata. As shown in tables S5 and S6 the BESTest indeed finds that this is the case, with a handful of exceptions, again confirming previous results that used less sophisticated techniques (*41*).

table S5. Malaria: BESTest P values for CP group metadata.

	Model			
Network	SBM	DCSBM	Poi. SBM	Poi. DCSBM
Malaria 1	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 2	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 3	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 4	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 5	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 6	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 7	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 8	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 9	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$

table S6. Malaria: BESTest P values for UPS metadata.

	Model			
Network	SBM	DCSBM	Poi. SBM	Poi. DCSBM
Malaria 1	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 2	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 3	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 4	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 5	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 6	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 7	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 8	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$
Malaria 9	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$	$< 10^{-5}$

C No optimal community detection algorithm

In the main text we argue that that the goal of recovering ground truth communities is ill posed because it lacks a unique solution and we also claim a "No Free Lunch" theorem for community detection. In the Supplementary Text, we describe and expound those claims using technical arguments.

For convenience, we provide a reference table of notation used in derivations in the Supplementary Text.

table S7. Notation used in the Supplementary Text.

Variable	Definition
$\mathcal{G}$	a network, $\mathcal{G} = (V, E)$
N	the number of nodes $\lvert V \rvert$
$\mathcal{T}$	ground truth (planted) partition
$\mathcal{C}$	detected communities partition
g	generative model, maps a partition to a network. $\mathcal{G} = g(\mathcal{T})$
f	comm. detection method, maps $\mathcal{G}$ to a partition $\mathcal{C} = f(\mathcal{G})$
$\ell(\cdot, \cdot)$	an error or loss function, returns a scalar
X	the space of possible inputs, i.e., all possible graphs $\mathcal{G}$
Y	the space of possible outputs, i.e., all possible partitions
γ	the true relationship between X and Y
h	the hypothesis about the relationship between X and Y
σ_X	probability density over X
$\Lambda(\ell)$	total loss across all possible inputs for loss function ℓ
u, v	two partitions of N objects
Ω	the set of all possible partitions of N objects.
$\mathcal{B}_N$	the Nth Bell number, the number of possible ways that N objects can be partitioned. $\mathcal{B} = \lvert\Omega\rvert$

C.1 Ground-truth community detection is an ill-posed inverse problem

A problem that is well posed satisfies three properties: (i) a solution exists, (ii) the solution is unique, and (iii) the solution's behavior changes continuously with the problem's initial conditions. The classic example of an ill-posed problem is the inverse heat equation, which violates condition (iii) because its solution (the distribution of temperature in the past) is highly sensitive to changes in the distribution of temperatures at the present. The problem of reproducing ground truth communities $\mathcal{T}$ from a network $\mathcal{G}$ by formulating the correct community detection

algorithm f^* is ill posed because it fails condition (ii), i.e., community detection has no unique solution.

> **Definition**: The *ground truth community detection problem*: given a fixed network $\mathcal{G}$ in which there has been hidden some ground truth partition $\mathcal{T}$, find the true communities that were planted in, embedded in, or used to generate the network. In other words, given $\mathcal{G}$, find the $\mathcal{T}$ such that $\mathcal{G} = g(\mathcal{T})$.

We now argue that the ground truth community detection problem is ill posed because its solution is not unique. The intuition behind this argument is that any network $\mathcal{G}$ could have been created using many different planted partitions via different generative processes. Therefore, searching for the ground truth partition without knowing the exact generative mechanism is an impossible task; there is no ground truth.

> **Theorem 1**: For a fixed network $\mathcal{G}$, the solution to the ground truth community detection problem is not unique.

Proof: We first show that the graph $\mathcal{G}$ can be produced by using two different planted partitions, $\mathcal{T}_1$ and $\mathcal{T}_2$ with $\mathcal{T}_1 \neq \mathcal{T}_2$. Let $\mathcal{T}_1$ be the trivial 1-partition in which all vertices are in the same group, and let g_1 be the generative model of Erdős-Rényi random graphs with probability $p \in (0, 1)$. Then the model and partition $g_1(\mathcal{T}_1)$ create $\mathcal{G}$ with non-zero probability. Let $\mathcal{T}_2$ be the trivial N-partition in which each vertex is in its own group, and let g_2 be a generative model that specifies the exact number of edges between all groups, such that $g_2(\mathcal{T}_2)$ produces $\mathcal{G}$ with probability one. We therefore have two partitions $\mathcal{T}_1 \neq \mathcal{T}_2$ and both $g_1(\mathcal{T}_1)$ and $g_2(\mathcal{T}_2)$ can create $\mathcal{G}$. Since two different planted partitions may be responsible for $\mathcal{G}$, both are potential solutions of the community detection problem. Therefore, the solution to the community detection problem is not unique for the network $\mathcal{G}$. To complete the proof, note that the 1-partition and N-partition argument above applies equally well to any network $\mathcal{G}$. □

The theorem above relies on two trivial partitions, the 1-partition and the N-partition in its proof, but other examples exist as well: consider the generative model $g_{\mathcal{G}^*}$ which maps any partition that it is given to some fixed network $\mathcal{G}^*$, essentially ignoring the information provided by the partition [similar to case (i) in the main text]. These models, while sufficient for the proof, are not particularly interesting for practictioners, yet non-trivial models and partitions also exist for any network. For instance, the Karate Club network may have plausibly been produced by the SBM with a core-periphery partition or by the degree-corrected SBM with a social faction partition (*20*).

Belief in ground truth $\mathcal{T}$ necessitates a belief in a specific generative mechanism g which together produced the network $\mathcal{G}$. For real-world networks, which may arise through more complex processes than those described here, we do not know the generative mechanism. Theorem 1 means that, in these cases, it is impossible to recover the *true* partition because *any* partition may plausibly have been used to generate the network. Therefore the ground truth community detection problem is ill-posed for any network for which the generative process is unknown because there is no unique solution. Put differently, it is impossible to uniquely solve an inverse problem when the function to be inverted is not a bijection.

C.2 No Free Lunch for community detection

The "no free lunch" (NFL) theorem (*32*) for machine learning states that for supervised learning problems, the expected misclassification rate, summed over all possible data sets, is independent of the algorithm used. In other words, averaged over all problems, every algorithm has the same performance. Therefore, if algorithm f_1 outperforms algorithm f_2 for one set of problems, then there exists some other set of problems for which algorithm f_2 outperforms algorithm f_1. In other words, it is impossible to get overall better performance without some cost; there is no free lunch.

The NFL theorem holds for community detection, and clustering problems in general. Demonstrating this requires that we first translate the community detection problem into the language and notation of the Extended Bayesian Framework (EBF) used in the NFL theorems for supervised learning. Then, under an appropriate choice of error (or "loss") function ℓ, the performance of any community detection method f, summed over all problems $\{g, \mathcal{T}\}$, is identical

$$\sum_{g,\mathcal{T}} \ell\big(\mathcal{T}, f\left(g(\mathcal{T})\right)\big) = \Lambda\ \ell) \quad \forall f \tag{27}$$

where $\Lambda(\ell)$ depends on the particular error function ℓ but is otherwise a constant, representing the total error.

In the following, we map community detection notation to EBF notation, provide a guiding example, and then resolve a subtle issue related to the loss function ℓ. We then discuss the implications of this result for future studies of community detection. The proofs of the NFL theorems are not recapitulated here, but are fully detailed in (*32*) and discussed extensively elsewhere.

C.2.1 Community detection in the Extended Bayesian Framework

The Extended Bayesian Framework (EBF) is a framework—a set of variables, definitions, and assumptions—for supervised learning that provides a clear and precise description of the problem. It is important in both the proof and implications of the NFL theorem, and was formalized at length in (*32*). In what follows, random variables will be denoted by capital letters, e.g. X, while instances of random variables will be denoted by the corresponding lowercase letters, e.g. x. In the EBF, we suppose that there exists an input space X, an output space Y, and that each of these has a countable (but possibly infinite) number of elements, $|X| = n$ and $|Y| = r$. The fundamental relationship to be learned is how X and Y are related, and to that end, let γ be the true or target relationship between X and Y, i.e., γ is the conditional distribution of Y,

given X. The points in the space X need not be distributed uniformly either, so we also specify σ, the probability density function of points x in the input space X, i.e., $P(x|\sigma) = \sigma_X$. In the nomenclature of community detection, the input $x \in X$ is simply the observed graph $\mathcal{G}$, and the output $y \in Y$ is the true partition into communities $\mathcal{T}$ for the nodes described by x. To solve a community detection problem, we hope to predict the true communities y from the input graph x; a community detection method will be successful when its hypothesized relationship h is an accurate representation of the true relationship γ between X and Y.

In supervised learning, for which the NFL theorems were originally proved, we aim to learn the relationship between X and Y from a training set d which consists of m ordered pairs of samples from X and Y, $\{d_X(i), d_Y(i)\}_{i=1}^{m}$. In response to the training data, the learning algorithm produces a hypothesis h in the form of an x-conditioned probability distribution over values y. The way in which the learning algorithm produces a hypothesis from training sets is described by $P(h|d)$, the distribution over hypotheses conditioned on the observed data. Note that the algorithm learns from the data alone and is independent of γ, i.e., $P(h|d, f) = P(h|d)$. If the algorithm performs well the hypothesis h will have high correspondence with the true relationship γ. Therefore, in supervised learning, algorithms are evaluated by their ability to make sufficient use of a limited training set to provide good predictions of y given x *not* in the training set. On the other hand, in unsupervised learning—a category which includes clustering and community detection—the training set d is empty ($m = 0$), so the prediction h is based solely on the prior beliefs encoded in the model $P(h)$. We note that in the NFL theorems for supervised learning, the independence of training data d from γ and σ is important to establish, but for unsupervised tasks, the set d is empty so it is trivially independent of γ and σ.

To better understand the EBF for community detection, an example is helpful. Consider the problem of finding two planted communities in a network $\mathcal{G}$. The true relationship γ between the network and its partition is hidden. Given only $\mathcal{G}$—which is a point in the space of

graphs X—fitting the parameters of an SBM, maximizing modularity, or using another method of our choice, produces a hypothesis h, which is a prediction about which nodes belong to which groups. If these communities are found correctly by the algorithm, then h will be highly correlated with the true communities mapped by γ. (This is equally true for both hard partitions, where each node belongs to only one group, and soft partitions, where each node may be distributed over multiple groups.) In other words, h estimates γ based on a point in X called $\mathcal{G}$. Because the estimate h is based *only* on $\mathcal{G}$ and the assumptions of the algorithm $P(h)$, it reproduces γ with possibly limited accuracy, and therefore its community assignments may or may not be highly correlated with the true assignments $\mathcal{T} \in Y$. Increasing the size of the input data set may help with accuracy as well: by generating a larger graph using the same generative model, $\mathcal{G}$ supplies a different point in X providing more information to the community detection method. This may allow the estimate h to produce better predictions of γ, thereby producing a more accurate partitioning of nodes into their true communities, but only if the model $P(h)$ is sufficiently aligned to reality $P(\gamma)$.

All learning algorithms make some prior assumptions, in the form of $P(h)$, about the possible relationships between inputs and outputs. For unsupervised methods such as community detection, there is a much greater importance associated with these assumptions because they do not have access to training data. For instance, a supervised algorithm could supposedly start from a uniformly ignorant prior $P(h)$ and rely on having a sufficiently large training set that $P(h|d)$ is informative. When there is no training data it is necessary that $P(h)$ is informative of the possible input-output relationship. Thus, community detection algorithms encode beliefs or definitions of community structure, and these beliefs constitute a prior over the kinds of problems that we expect to see. Some methods, for example, search only for assortative (*9, 37*) or disassortative (*38*) community structures, while other are more flexible and can find mixtures of assortative, disassortative, and core-periphery structures (*15, 16, 20, 39*) and allow for nodes

to belong to multiple communities (*36, 37*).

C.2.2 Loss functions and *a priori* superiority

So far, we have discussed the phrasing of community detection in the language of EBF but have not described the way in which error (also called loss or cost) is measured. The error function quantifies the accuracy of predictions, and the EBF introduces a random variable C which represents the error associated with a particular γ and h, i.e., the error associated with using a particular algorithm for a particular problem. Conceptually, this is what the community detection literature attempts to estimate when algorithms are compared based on their ability to recover planted communities in synthetic data. More formally, C is measured by the distribution $P(c|h,\gamma,d\,)$, which incorporates the relationships between the test set and the generating process, as well as the way in which the hypothesis is related to the training data. Therefore, the quantity of interest to those developing algorithms is the expected error, $E(C|h,\gamma,d\,)$. For example, in the context of supervised learning, choosing the loss function ℓ to be the average misclassification rate is common. For the purposes of community detection, misclassification rate is not of interest for a pedantic but important reason: for community detection and other related unsupervised tasks such as clustering, permutations of the group labels are inconsequential because the partition is the desired outcome; labeling two groups a and b is equivalent to labeling them b and a. As a result, many of the loss functions typically used to compare partitions have a "geometric" structure that implies an *a priori* superiority of some algorithms, which would appear to contradict the NFL theorem. We now discuss one such loss function frequently used to evaluate community detection algorithms, the normalized mutual information, and the structure that it imposes on the space of partitions.

Normalized mutual information is an information-theoretic measurement of similarity between two partitions that treats both partitions as statistical objects. For a partition u of N

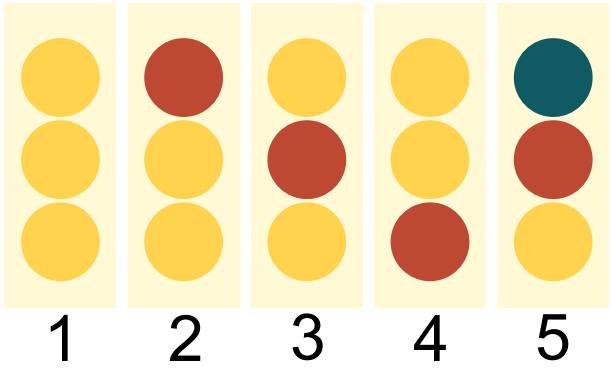

fig. S6. The five distinct ways to partition three nodes. Normalized mutual information and adjusted mutual information between each pair of partitions are presented in tables S8 and S9, respectively.

objects into K_u groups, the probability that an object chosen uniformly at random falls into group u_i is $p_i = |u_i|/N$, $i = 1 \dots K_u$. The entropy associate with a partition u is then the entropy of its corresponding distribution p

$$H(u) = -\sum_{i=1}^{K_u} p_i \log\left(p_i\right)$$

When comparing two partitions u and v of the same set of objects, each object belongs to some group u_i in the first partitions and some other group v_j, $j = 1 \dots K_v$ in the second partition, with the corresponding probability p_{ij}. The mutual information between the two partitions is therefore

$$I(u, v) = \sum_{i=1}^{K_u} \sum_{j=1}^{K_v} p_{ij} \log\left(\frac{p_{ij}}{p_i p_j}\right)$$

which can be normalized to define normalized mutual information as

$$\text{NMI}(u, v) = \frac{I(u, v)}{\sqrt{H(u)H(v)}} \tag{28}$$

Other normalizing factors in the denominator are possible, including $\frac{1}{2}[H(u) + H(v)]$ and $\max\{H(u), H(v)\}$; see (*52*). NMI maps partitions to the unit interval, with 0 indicating that two partitions are uncorrelated and 1 indicating that they are identical (even if the groups labels differ).

To understand how an error function imposes a geometric structure, consider a simple problem (unrelated to community detection) of predicting, based on some inputs X, a point in the unit circle in $Y = \left\{ y \;\middle|\; \|y\| \leq 1,\ y \in \mathcal{R}^2 \right\}$. If all points in Y are equally likely, then an algorithm that guesses the center of the circle $h = 0$ will outperform an algorithm that guesses a point on the boundary $h \in \partial Y$, simply due to the fact that the center of the circle is, on average, closer to the other points of the circle than any boundary point. Normalized mutual information imposes a geometric structure on the space of partitions in a similar way.

Consider a loss function based on normalized mutual information (NMI) and imagine a community detection algorithm that entirely ignores the network and simply returns a fixed partition of the vertices. As in the example above, NMI provides a geometrical structure on the space of partitions, an algorithm that always returns a partition toward the middle of the space of partitions will outperform an algorithm that always returns a partition on the boundary of that space. To demonstrate this point, fig. S6 shows all five possible partitions of three vertices, and table S8 shows their NMI for all pairwise comparisons. Averaged over all possible correct answers, an algorithm that consistently predicts partition 5 will outperform all others, and an algorithm that consistently predicts partition 1 will underperform all others. However, this structure is a known issue of NMI, and so other error functions and corrections have been proposed such as the adjusted mutual information (AMI), which accounts for the geometry of the space (*52*). Table S9 shows the AMI for the same set of partitions, and the expected AMI is zero except for the partition that contains only a single group and the partition of each node into separate groups. In the case of these partitions, the 1-partition and the N-partition, the expected AMI is the reciprocal of the Bell number $\mathcal{B}_N$—the Bell number is the total number of distinct ways that N objects can be partitioned, and it grows superexponentially with N—so as the number of vertices N increases, so AMI approaches 0 superexponentially; for even small networks, $1/\mathcal{B}_N \approx 0$. In this way, AMI provides a "geometry-free" space in which no

table S8. Normalized mutual information for partitions in fig. S6.

	Partition 2				
Partition 1	**1**	**2**	**3**	**4**	**5**
1	1	0	0	0	0
2	0	1	0.27	0.27	0.76
3	0	0.27	1	0.27	0.76
4	0	0.27	0.27	1	0.76
5	0	0.76	0.76	0.76	1
$\mathbb{E}$[NMI]	0.20	0.46	0.46	0.46	0.66

table S9. Adjusted mutual information for partitions in fig. S6.

	Partition 2				
Partition 1	**1**	**2**	**3**	**4**	**5**
1	1	0	0	0	0
2	0	1	-0.5	-0.5	0
3	0	-0.5	1	-0.5	0
4	0	-0.5	-0.5	1	0
5	0	0	0	0	1
$\mathbb{E}$[AMI]	0.20	0	0	0	0.20

one partition is *a priori* closer to all others. This key property of AMI, called homogeneity, is proved in a Lemma in the next section.

C.2.3 Lemma and theorems

We now prove a lemma about adjusted mutual information, and then formally state the NFL theorem for supervised learning and prove the no free lunch theorem for community detection.

> **Lemma 1**: Adjusted mutual information (AMI) is a homogenous loss function over the interior of the space of partitions of N objects. Including the boundary partitions, i.e., the 1-partition and the N-partition, AMI is homogenous within $\mathcal{B}_N^{-1}$.

Proof: Showing that AMI is a homogenous loss function requires that we show

$$L(u) = \sum_{v \in \Omega} \text{AMI}(u, v) \tag{29}$$

is independent of u, where Ω is the space of all partitions of N objects. Stated plainly, if $L(u)$ is independent of u, it means that the total AMI between partition u and all possible partitions will be the same, no matter which partition u is chosen. The definition of AMI is

$$\text{AMI}(u, v) = \frac{I(u, v) - E[I(u, v)]}{\sqrt{H(u)H(v)} - E[I(u, v)]}$$

where I is mutual information and H is entropy (*52*). The AMI takes on a value of 1 when two partitions are identical and a value of 0 when they are only correlated to the extent that one would expect by chance. In particular, the expectation E is taken over all possible pairs of partitions u' and v' such that every u' has the same number of groups and the same number of objects belonging to each group as does u, and likewise for v' and v. In this way, the expectation E is taken over all pairs of divisions that preserve the group sizes of the two partitions being compared. For convenience of notation, let ϕ be a subset of all partitions Ω such that every partition $v \in \phi$ has the same number of groups and same number of objects in each group. The set of all partitions Ω may be subdivided into non-overlapping subsets $\{\phi_i\}$, such that $\cup_i \phi_i = \Omega$ and $\phi_i \cap \phi_j = \varnothing$ for any $i \neq j$. (For example, in fig. S6, partition 1 belongs to ϕ_1, partitions 2, 3, and 4 belong to ϕ_2, and partition 5 belongs to ϕ_3.) Let the particular subset ϕ_i to which a partition u belongs be denoted by $\phi(u)$.

Prior to proceeding, we note that there are two special boundary partitions, the 1-partition in which all objects are in a single group and the N-partition in which each object is in its own group. These will be denoted by $\bar{1}$ and $\bar{N}$ respectively. Note that $\bar{1} = \phi(\bar{1})$ so that $|\phi(\bar{1})| = 1$, and that $\phi(\bar{N})$ is equivalent to the set of all possible relabelings of the N objects, so that $|\phi(\bar{N})| = N!$. Because there is only one element of $\phi(\bar{1})$, it is necessarily

true that $I(\bar{1},\bar{1}) = E[I(\bar{1},\bar{1})] = H(\bar{1})$. Thus, for this special case, the numerator and denominator of AMI are identical, and AMI$(\bar{1},\bar{1}) = 1$. Similarly, because the set $\phi(\bar{N})$ contains every possible permutation of the labels of the objects, yet all partitions are identical, $I(\bar{N},\bar{N}) = E[I(\bar{N},\bar{N})] = H(\bar{N})$, and so AMI$(\bar{N},\bar{N}) = 1$.

In order to prove Eq. (29), we will show that $L(u) = 0$ for all u except $\bar{1}$ and $\bar{N}$ by demonstrating that the numerator of the definition of AMI is 0, specifically

$$\sum_{v\in\Omega} [I(u,v) - E[I(u,v)]] = 0 \;\; \forall\, u \neq \bar{1} \text{ or } \bar{N} \tag{30}$$

In fact, we will show that Eq. (30) holds by breaking the entire sum over all partitions Ω into sums over each of its disjoint subsets $\{\phi_i\}$, and proving that

$$\begin{aligned} &\sum_{v'\in\phi(v)} [I(u,v') - E[I(u,v')]] = 0 \\ &\forall\, u \text{ and } \forall\, v \text{ except } u = v = \bar{1} \text{ or } u = v = \bar{N} \end{aligned} \tag{31}$$

In other words, we will show that the numerator of the definition of AMI is equal to zero when summed over any subset $\phi(v)$ for any fixed partition u, except the boundary cases that both u and v are equal to $\bar{1}$ or both are equal to $\bar{N}$. We first examine the expectation term in Eq. (31). Recall that the expectation is taken over all pairs of members of the subsets $\phi(u)$ and $\phi(v)$, respectively

$$E[I(u,v)] = \frac{1}{|\phi(u)||\phi(v)|} \sum_{u'\in\phi(u)} \sum_{v'\in\phi(v)} I(u',v') \tag{32}$$

In fact, because the sums above are taken over the subsets $\phi(u)$ and $\phi(v)$ that contain u and v, the expected mutual information is equal to a constant ζ for any pair of partitions drawn from $\phi(u)$ and $\phi(v)$

$$E[I(u,v)] = \zeta \quad \forall\, u \in \phi(u) \text{ and } \forall\, v \in \phi(v) \tag{33}$$

Note then that we may rewrite the sum over expectations in Eq. (31) as $\sum_{v'\in\phi(v)} E[I(u,v')] = |\phi(v)|\;\zeta$. Therefore, it remains to be shown that the sum over mutual informations in Eq. (31)

is also equal to $|\phi(v)|\,\zeta$

$$\sum_{v' \in \phi(v)} I(u, v') = |\phi(v)|\,\zeta \tag{34}$$

To see that this is true, despite the fact that u is fixed (and not averaged over all $u' \in \phi(u)$ as in $E[I(u, v)]$), note that Eq. (34) nevertheless sums over every $v' \in \phi(v)$ which is the set of every randomization of the partition v, provided group sizes are held constant. Because this includes all relabelings (or reindexings) of the N objects being partitioned, it must be true that

$$\sum_{v' \in \phi(v)} I(u_1, v') = \sum_{v' \in \phi(v)} I(u_2, v') \text{ whenever } u_1 \in \phi(u_2) \tag{35}$$

In other words, the sum of mutual information between a fixed partition u_1 and all members of a subset $\phi(v)$ must be equal to the sum of mutual information between a different fixed partition u_2 and the same subset $\phi(v)$, but only if u_1 and u_2 both belong to the same subset as each other. Therefore, Eq. (34) is true, meaning that the sum over the two terms in Eq. (31) is zero, independent of u. This first implies that the AMI between any boundary partition and any interior partition is 0, $\text{AMI}(u, \bar{1}) = 0$ for any $u \neq \bar{1}$ and $\text{AMI}(u, \bar{N}) = 0$ for any $u \neq \bar{N}$. This, in turn, implies Eq. (30) is true. This completes the proof of the first statement, that Eq. (29) is true, and in particular, $L(u) = 0$, for any $u \neq \bar{1}, \bar{N}$ and AMI is homogeneous over all non-boundary partitions.

In the special cases of $u = v = \bar{1}$ and $u = v = \bar{N}$, note that we have already shown that $\text{AMI}(\bar{1}, \bar{1}) = 1$, $\text{AMI}(\bar{N}, \bar{N}) = 1$, and $\text{AMI}(u, \bar{1}) = 0$ for any $u \neq \bar{1}$ and $\text{AMI}(u, \bar{N}) = 0$ for any $u \neq \bar{N}$. Therefore

$$\begin{aligned} L(\bar{1}) &= \sum_{v \in \Omega} \text{AMI}(\bar{1}, v) = \mathcal{B}_N^{-1} \\ L(\bar{N}) &= \sum_{v \in \Omega} \text{AMI}(\bar{N}, v) = \mathcal{B}_N^{-1} \end{aligned} \tag{36}$$

completing the proof of the second statement: including the boundary points, AMI is homogenous within an additive constant $\mathcal{B}_N^{-1}$. □

Theorem 2 (Wolpert 1996): For homogeneous loss ℓ, the uniform average over all γ of $P(c|\gamma, d)$ equals $\Lambda(c)/r$.

Proof: See (*32*).

Theorem 3 (No free lunch for community detection): For the community detection problem with a loss function of adjusted mutual information, the uniform average over all γ of $P(c|\gamma)$ equals $\Lambda(c)/r$.

Proof: Lemma 1 proves that adjusted mutual information is homogeneous and applying Theorem 2 with $d = \varnothing$ completes the proof. □

C.2.4 Implications

No free lunch for community detection means that, uniformly averaged over all community detection problems, and evaluated by AMI, all algorithms have equivalent performance. Phrased more usefully, it means that any subset of problems for which an algorithm outperforms others is balanced by another subset for which the algorithm underperforms others. Thus, there is no single community detection algorithm that is best overall.

On the other hand, if the set of problems of interest is a non-uniform subset of all problems, then one algorithm may outperform another on this subset. In other words, the bias of an algorithm to solving a particular type of community detection problem may be its strength, accepting the fact that such an advantage must be balanced by disadvantages elsewhere. For instance, algorithms like the unconstrained SBM (which can find both assortative and disassortative communities and mixtures and gradations thereof) are not universally superior to versions of the SBM constrained to find only assortative or disassortative communities (*38*)—if the particular subset of problems is believed to contain only disassortative communities, then the unconstrained SBM will not perform as well as a constrained one. In other words, no free lunch for community detection means that matching the assumptions in the model to the under-

lying generative process can lead to better, more accurate results, but only in the cases when the beliefs about the underlying generative process are accurate; in the other cases, the same model assumptions that improved performance on some problems will diminish it for others. To some extent we expect the distribution of problems to be non-uniform in general. Out of all the possible ways of constructing a graph there may be some types of graph we are less likely to observe. For each graph we can also expect that of all the possible partitions, many will correspond to random assignments of nodes that are not useful in any application. Put differently, there may be some problems we do not wish to solve—but, unless we know which problems they are, it offers us little or no benefit in practice. We note that relatively little is known about which algorithms perform better than others within particular domains or on particular classes of networks. A valuable line of future research on community detection will be developing such an understanding (*49, 50*).

D Datasets and additional methodology

D.1 Lazega Lawyers networks

The Lazega Lawyers network is a multilayer network consisting of 71 attorneys of a law firm with three different sets of links, corresponding to friendships, exchange of professional advice, and shared cases (*51*). The original study also collected five sets of categorical node metadata, corresponding to status (partner or associate), gender, office location, type of practice (corporate or litigation), and law school (Harvard, Yale, UConn, other). The relationships and dynamics within the law firm were studied extensively in the initial publication of these data sets, but they were not primarily analyzed as complex networks.

D.2 Malaria *var* gene networks

The Malaria data set consists of 307 *var* gene sequences from the malaria parasite *P. falciparum* (*41*). Each *var* gene encodes a protein that the parasite uses to evade the human immune system, and therefore this family of genes is under intense evolutionary pressures from the human host. The original study focused on uncovering the functional and evolutionary constraints on *var* gene evolution by identifying community structure in *var* gene networks.

These sequences were independently analyzed at 9 loci (locations within the genes), producing 9 different genetic-substring-sharing networks with the same node set. In other words, there are 9 layers in this multilayer network. Each parasite genome contains around 60 *var* genes, and the 307 genes in this data set represent seven parasite genomes. The original study included three sets of categorical node metadata, corresponding to the upstream promoter sequence classification (UPS, $K = 3$), CysPoLV groups (CP $K = 6$), and the parasite genome from which sequence was generated (parasite origin $K = 7$).

D.3 Karate Club network

The Zachary Karate Club represents the observed social interactions of 34 members of a karate club (*14*). At the time of study, the club fell into a political dispute and split into two factions, which are treated as metadata. The Karate Club has been analyzed exhaustively in studies of community detection, and its faction metadata have often been used as ground truth for community detection, due to the network's small size and easily interpretable social narrative.

D.4 Generation of log-likelihood surface plots

The log-likelihood surface plots in Figs. 1, 3, 4, S1, S3, and S4 illustrate the changes in log likelihood as the partition of network nodes is varied. In the figures, we show surfaces that appear to be continuous over that two dimensional space, in spite of the fact that the true space of partitions is high dimensional and discretized, and so here we explain the methods used to produce visually meaningful plots.

Plots were generated in three steps: partition sampling, data projection and surface interpolation. For most networks it is infeasible to calculate the log likelihood of all possible partitions, so we instead sampled a subset of partitions. We began with the set of partitions along the path of the neoSBM (e.g., Fig. 3) and sampled partitions around the local neighborhood of this initial set. Specifically, we did so by selecting two partitions uniformly at random from the initial set and created each new partition by assigning q nodes (chosen randomly and uniformly) to the group assignment of the first partition and the remaining $N - q$ nodes to that of the second partition.

Next, we projected the K^N-dimensional partition data down to two dimensions using Multi-dimensional Scaling (MDS) (*59*) and variation of information (*60*) as a similarity measure. The outcome of this projection was a two-dimensional representation of the partition space that preserves the variation of information between partitions.

Finally, we used MATLAB's *scatteredInterpolant* function with *natural* interpolation to fit an interpolated surface to the data, which we evaluated over a grid of domain points and smoothed using a Gaussian kernel to improve legibility. The processes of embedding, interpolating, and smoothing are not particularly sensitive to changes in parameters or grid resolutions. In the special case of Fig. 4, we also plotted the partitions of the neoSBM in addition to the interpolated log-likelihood surface to illustrate the neoSBM's path in the broader context of the surface. There were no modifications or smoothing of the points of the neoSBM's path beyond the embedding process described above.